Rechnerarchitektur

Aufgabe 1 – Zahlendarstellung und Rechnung

1

Warum werden i. d. R. Fließkommazahlen und nicht Festkommazahlen zur Darstellung von reellen Zahlen verwendet? Anders formuliert: Welche Vorteile bringt diese Art der Darstellung?

- ➔ Man kann sehr große und sehr kleiner Zahlen darstellen
- ➔ Z.B. “reelle Zahl“ * 10^23
- ➔ Oder auch „reelle Zahl“ * 10^-23

2

Für eine Fließkommazahl-Darstellung werden die gegebenen Bits in drei Abschnitte unterteilt. Wie heißen diese Abschnitte?

- ➔ Vorzeichenbit (0 / 1)
- ➔ Charakteristik - Exponent
- ➔ Significant - Mantisse

3

Welchen Vorteil bringt es, wenn dem Exponenten mehr Bits zugeteilt werden bzw. welchen Vorteil bringt es, wenn der Mantisse mehr Bits zugeteilt werden?

- ➔ Je mehr Bits der Mantisse zugeordnet sind, desto genauer kann eine Zahl sein, weil die Mantisse die Genauigkeit einer Zahl bestimmt.
- ➔ Aber, wenn man große Zahlen darstellen möchte, dann muss man dem Exponenten mehr Bit zuteilen, da der Exponent den darstellbaren Zahlenbereich vorgibt.

4

Erklären Sie Unterlauf (Underflow) und Überlauf (Overflow) für Fließkommazahlen.

- ➔ Ein Underflow bedeutet; ich rutsche mit meiner Zahl in eine Zahl rein, die kleiner ist als die kleinste darstellbare Zahl.
- ➔ Wenn ich Betragsmäßig über die größte Zahl gehe, sowohl für positive als auch für negative Zahlen, dann hab ich einen Überlauf

5

Nennen Sie Beispiele für Festlegungen, die der IEEE-754 Standard mitbringt. Warum ist eine solche Standardisierung sinnvoll?

- ➔ Es gibt sehr viele verschiedene Darstellungsformen von Binärzahlen.
- ➔ Das führte dazu, dass je nach Hersteller anders gerechnet wurde.
- ➔ Deshalb hat man einen Standard eingeführt, den IEEE-754.
  Den Standard befolgen alle Systeme heute und haben in ihren Rechnungen, dieselben Abweichungen.
- ➔ Dieser Standard kann sicherstelle, dass Zahlen richtig gerundet und Ausnahmen, z.B. durch 0 dividieren oder Überlauf, gut behandelt werden.

6

Was ist die betragsmäßig größte bzw. kleinste darstellbare Zahl im IEEE-754 32 Bit Standard?

➔ Größte Zahl: $(2-2^{-23})*2^{127}$

➔ Kleinste Zahl: $1 * 2^{-126}$.

➔ Die größte Zahl hat die größste Mantisse und Exponent: $1.11111111111111111111111 * 2^{127}$.

➔ Die kleinste Zahl hat die kleinste Mantisse und Exponent: $1.00000000000000000000000 * 2^{-126}$.

Zahlen in IEEE-754 32 bit Darstellung rechnen

1) -592.183940 + 0.91213

-592.183940 in IEEE-754:

Vorzeichen: 1

592 / 2 = 296 rest 0

296 / 2 = 148 rest 0

148 / 2 = 74 rest 0

74 / 2 = 37 rest 0

37 / 2 = 18 rest 1

18 / 2 = 9 rest 0

9 / 2 = 4 rest 1

4 / 2 = 2 rest 0

2 / 2 = 1 rest 0

1 / 2 = 0 rest 1

592 = 100101000

0.18394 * 2 = 0.36788

0.36788 * 2 = 0.73576

0.73576 * 2 = 1.47152

0.47152 * 2 = 0.94304

0.94304 * 2 = 1.88608

0.88608 * 2 = 1.77216

0.18394 = 0.00101111000101101011

Binär: $1001010000.00101111000101101011 * 2^0$

Normalizieren: $1.001010000000101111000101101011 * 2^9 \approx 1.00101000000101111000110 * 2^9$ (Round-up)

Mantisse: 00101000000101111000110

Charakteristik: $9 + 127 = 136 = 10001000_2$

Ergebnis: 1 10001000 00101000000101111000110

0.91213 in IEEE-754:

Vorzeichen: 0

Binär: $0.11101001100000001011 * 2^0$

Normalizieren: $1.1101001100000001011 * 2^{-1}$

Mantisse: 11010011000000010110000

Charaktieristik: $-1 + 127 = 126 = 1111110_2$

Ergebnis: 0 1111110 11010011000000010110000

Addition:

Denormalizieren: -1.00101000000101111000101101011 * 2^{9} + 1.1101001100000001011 * 2^{-1} = 1.00101000000101111000101101011 * 2^{9} - 0.000000000111010011000000001011 * 2^{9}

Addieren:

```
    1.00101000000101111000101101011
 -  0.00000000011101001100000010110
    1.00100111101000101100101010101
```

Vorzeichen: 1

Mantisse: 00100111101000101100101010101

Charakteristik: 9 + 127 = 136 = 10001000

Ergebnis: 1 10001000 00100111101000101100 11

2) 3981.1729 * -2.91762

3981.1729 in IEEE-754:

Vorzeichen: 0

Binär: 111110001101.00101100010000110011 * 2^{0}

Normalizieren: 1.11110001101001011000100001100 11 * 2^{11}

Mantisse: 11110001101001011000 10

Charaktieristik: 11 + 127 = 138 = 10001010

Ergebnis: 0 10001010 11110001101001011000 10

3981 / 2 = 1990 rest 1

1990 / 2 = 995 rest 0

995 / 2 = 497 rest 1

497 / 2 = 248 rest 1

248 / 2 = 124 rest 0

124 / 2 = 62 rest 0

62 / 2 = 31 rest 0

31 / 2 = 15 rest 1

15 / 2 = 7 rest 1

7 / 2 = 3 rest 1

3 / 2 = 1 rest 1

1 / 2 = 0 rest 1

-2.9176 in IEEE-754:

Vorzeichen: 1

Binär: 10.1110101011100111 1101 * 2^{0}

Normalizieren: 1.01110101011100111 1101 * 2^{1}

Mantisse: 0111010101110011111010

Charakteristik: 1 + 127 = 128 = 10000000

Ergebnis: 1 10000000 0111010101110011111010

Multiplizeren:

1.11110001101001011000100001100 11 * 2^{11} * 1.01110101011100111 1101 * 2^{1} =

10110010110000.0110100110101101 = 1.01100101100000110100110101101 * 2^{13}

Vorzeichen: 1

Mantisse: 01100101100000110100110101101

Charaktieristk: 13 + 127 = 140 = 10001100
Ergebnis: 1 10001100 01100101100000011010011